**Sample Refactoring Documentation for Project “Bulls and Cows”**

1. Redesigned the project structure: Team “Bulls-and-Cows-6”

* Renamed the solution from domashno to BullsAndCowsGame
* Renamed the project from domashno to BullsAndCowsGame
* Added Namespace BullsAndCowsGame to BullsAndCows class
* Added Namespace BullsAndCowsGame to ScoreBoard class
* Added new Unit Test Project

1. Renames

* Class: klasirane -> ScoreBoard
* File: klasirane.cs -> ScoreBoard.cs
* In class BullsAndCows field: digits -> secretDigits
* In class BullsAndCows field: helpExpression to helpDigits
* In class BullsAndCows variable: count3 -> bulls.
* In class BullsAndCows variable: count4 -> cows.
* In class BullsAndCows variable: count1 -> playerAttemps.
* In class BullsAndCows variable: flag1 -> gameFinished.
* In class BullsAndCows variable: r -> randomGenerator.
* In class BullsAndCows variable: count2 -> usedCheats.
* In class BullsAndCows const: ENTER\_GUES -> ENTER\_GUESS.
* In class BullsAndCows const: WRONG\_GUES -> WRONG\_GUESS.
* In class BullsAndCows const: WRONG\_INPUT -> WRONG\_INPUT\_FORMAT.
* In class BullsAndCows method: SetDigits -> SetNewDigits
* In class BullsAndCows method: ProccessGues -> CalculateBullsAndCows.
* In class ScoreBoard const: IN\_SCOREBOARD -> ASK\_NAME\_FOR\_SCOREBOARD.
* In class ScoreBoard const: OUT\_SCOREBOARD -> NOT\_FOR\_SCOREBOARD.
* In class ScoreBoard const: KLASIRANE\_RAZMER -> SCOREBOARD\_SIZE.
* In class ScoreBoard field: klasiraneto -> scoreBoard.
* In class ScoreBoard method: sort -> GetScoreBoardAsString
* In class ScoreBoard method: Sort -> SortScoreBoard

1. Reformatted the source code:

* Removed unused usings
* Moved all using inside of the namespace.
* Added new lines where needed according to High Quality Code practice.
* Removed all misplaced empty lines, e.g. in the method SetNewDigits or end of BullsAndCows.cs file
* Added **this.** prefix to all members of the class where needed
* Changed access modifier for all constants to public
* Added access modifier public to class BullsAndCows
* Removed useless comments
* Moved class constructors to the right place according to the best practices.
* Moved all private methods to bottom of the class.
* Refactored GetScoreBoardAsString method to return string instead of using Console class.
* Changed access modifier of method CalculateBullsAndCows from private to public
* Long line code break/refactored into several lines for easy read and debug, e.g.

|  |
| --- |
| this.scoreBoard.Add(new KeyValuePair<string, int>(name, attempts)); |
| var score = new KeyValuePair<string, int>(name, attempts);  this.scoreBoard.Add(score); |

* Replaced complicated logic in method SortScoreBoard of class ScoreBoard with new simple one.
* Removed code repetitions. e.g.

|  |  |
| --- | --- |
| string line = Console.WriteLine();  line.Trim().ToLower().CompareTo("top")  line.Trim().ToLower().CompareTo("restart") | string input = Console.ReadLine();  string line = input.Trim().ToLower();  if (line.CompareTo("top") == 0)  if (line.CompareTo("restart ") == 0) |

* Changed value of START\_EXPRESSION const to cover requrements
* Variables and fields made camelCase, types and methods made PascalCase.
* Added one more parameter to CalculateBullsAndCows method to make it testable
* Removed useless **new string('-', 80)** separator code printing from ProcessGuessedSecredDigits method of class BullsAndCows
* GetScoreBoardAsString method no longer calls SortScoreBoard method. The scoreboard gets sorted immediately after adding new record.
* GetScoreBoardAsString method now returns string instead of printing directly on the Console
* Replaced while loop with do-while loop in method Help of BullsAndCows class due to code repetition

1. Introduced constants:

* SCOREBOARD\_EMPTY = "Scoreboard is empty!"
* CONGRATULATIONS\_WITH\_CHEATS
* CONGRATULATIONS\_WITHOUT\_CHEATS
* SCOREBOARD\_TITLE = "Scoreboard:"
* SCOREBOARD\_INPUT\_FORMAT = "{0}. {1} --> {2} guesses"

1. Bug fixes

* Fixed a bug in CalculateBullsAndCows method in BullsAndCows class to cover requirement: “Note that a single digit can either represent a bull, or a cow, but not two bulls, two cows or bull and cow in the same time.” e.g.

|  |  |
| --- | --- |
| Secret: 2424, Guess: 4144 | |
| **Wrong** bulls: 1, cows: 2. | **Right** bulls: 1, cows: 1. |

* Fixed a bug where SortScoreBoard method of ScoreBoard class sorts the scoreboard on the opposite way.
* Fixed a bug in StartGame method of BullsAndCows class where the game always prints “Wrong input format!” after using “top” command

|  |  |
| --- | --- |
| Secret: 2424, Guess: 4144 | |
| top  Scoreboard empty!  **Wrong input format!** | top  Scoreboard:  1. Pesho --> 31 guesses  **Wrong input format!** |

1. Extracted private method ProcessGuessedSecredDigits from StartGame method.
2. Extracted private methods CalculateBulls and CalculateCows from CalculateBullsAndCows method.
3. Class ScoreBoard no more implements Singleton pattern. BullsAndCows class now use DI, Strategy pattern.

* Introduced interface IScoreBoard implemented by ScoreBoard class.

|  |
| --- |
| bool IsHighScore(int points);  void Add(string name, int points);  string GetScoreBoardAsString(); |

* Introduced private readonly field scoreboard of type IScoreBoard

1. The field randomGenerator of type Random is replaced with readonly field randomGenerator of type IRandomNumberProvider.

* Introduced class RandomNumberProvider.
* Introduced interface IRandomNumberProvider implemented by RandomNumberProvider class.

|  |
| --- |
| int GetRandomNumber(int minValue, int maxValue); |

1. Introduced constructor in BullsAndCows class asking for IRandomNumberProvider and IScoreBoard parameters. Plus Poor man’s DI